Università della Calabria

Dipartimento di Informatica, Modellistica, Elettronica e Sistematica

Corso di studio in

Ingegneria Elettronica

" Ottimizzazione del flusso energetico di un veicolo ibrido con architettura parallela"

Studente
Giuseppe Pirilli
Matricola: 237909

Anno Accademico 2023/2024

Indice

Introduzione

Questo progetto ha come obiettivo l'ottimizzazione del flusso di potenza tra le fonti energetiche (ICE ed EM) che costituiscono il powertrain di un HEV con architettura parallela su un orizzonte definito.

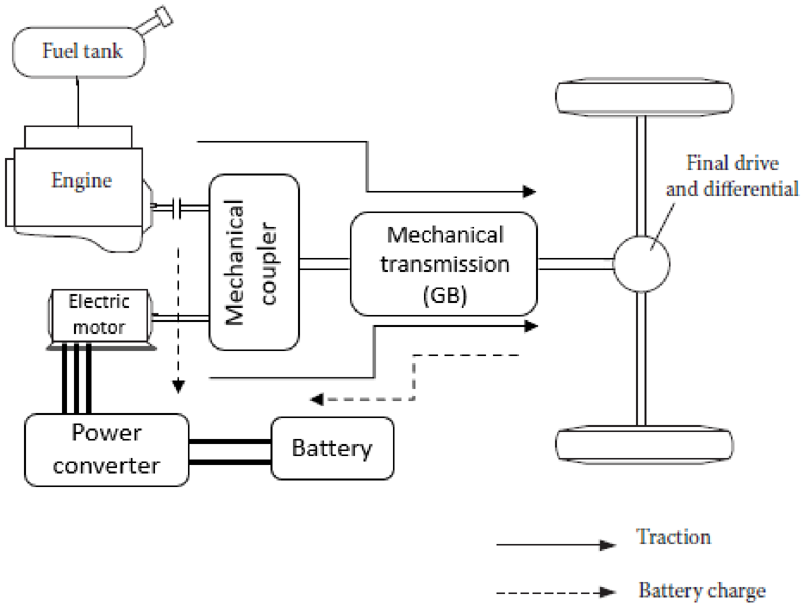
L'ottimizzazione consiste nel decidere la quantità di potenza fornita in ogni istante dalle fonti energetiche di bordo, in funzione della potenza richiesta, attraverso lo sviluppo di diverse strategie di controllo o di gestione dell'energia (EMSs).

L'intento è cercare un compromesso ottimale tra il consumo totale di carburante e la perdita di energia immagazzinata nella batteria (Charge-Sustaining) su un orizzonte di ottimizzazione definito soddisfacendo la potenza richiesta e rispettando tutti i vincoli operativi.

Le prestazioni delle strategie di controllo vengono testate e confrontate simulando cicli di guida standardizzati che emulano condizioni di guida differenti (urbano, extra-urbano, combinato).

Architettura e parametri HEV

— Architettura parallela

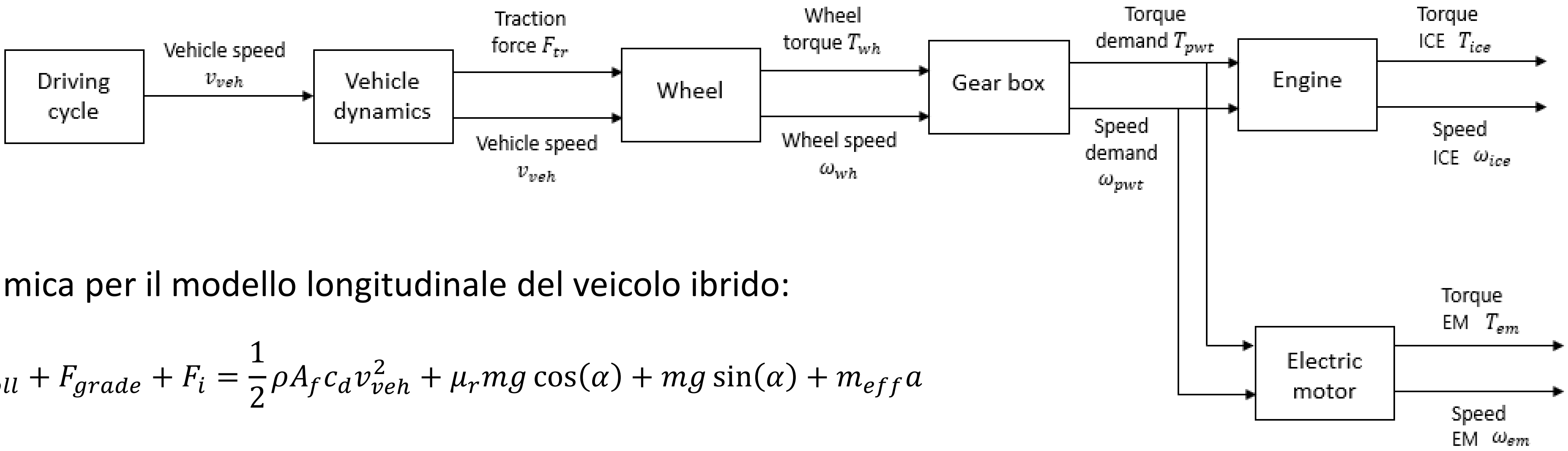


Hyundai Tucson Full-Hybrid

Powertrain	Parameter	Value
Vehicle	Vehicle mass (m)	$1649 + (3 \cdot 70)\ kg$
	Vehicle effective mass (m_{eff})	$1899\ kg$
	Frontal area (A_f)	$2.6\ m^2$
	Drag coefficient (c_d)	0.33
	Tire radius (r_w)	$0.35\ m$
	Rolling resistance coefficient (μ_r)	0.015
6-speed transmission (Gear box)	1st gear ratio $(i_{gb,1})$	4.639
	2nd gear ratio $(i_{gb,2})$	2.826
	3rd gear ratio $(i_{gb,3})$	1.841
	4th gear ratio $(i_{gb,4})$	1.386
	5th gear ratio $(i_{gb,5})$	1
	6th gear ratio $(i_{gb,6})$	0.772
	Final drive or differenzial gear ratio (i_{fd})	3.32
	Mechanical efficiency (η_{mech})	0.95
Engine	Maximum torque $(T_{ice,max})$	$265\ Nm$ @ $4500\ rpm$
	Maximum power $(P_{ice,max})$	132.2 kW @ $5500\ rpm$
Electric motor (Brushless PMSM)	Maximum torque $(T_{em,max})$	$264\ Nm$ @ $1600\ rpm$
	Maximum power $(P_{em,max})$	44.2 kW @ $1600\ rpm$
Battery pack (Li-ion)	Open circuit voltage $(V_{oc,nom})$	$270\ V$
	Capacity (Q_{nom})	$22\ Ah$
	Energy (E_{nom})	$5.94\ kWh$

Longitudinal dynamic modelling

Partendo da un ciclo di guida predefinito, è possibile conoscere in anticipo l'esatto profilo di potenza richiesta lungo tutta la missione di guida. Questo metodo è noto come 'backward approach' ed è necessario per poter applicare alcune strategie di controllo come: Dynamic programming (DP) o Pontryagin's minimum principle (PMP).



Equazione dinamica per il modello longitudinale del veicolo ibrido:

$$F_{tr} = F_{aero} + F_{roll} + F_{grade} + F_i = \frac{1}{2}\rho A_f c_d v_{veh}^2 + \mu_r mg\cos(\alpha) + mg\sin(\alpha) + m_{eff}a$$

Da cui:

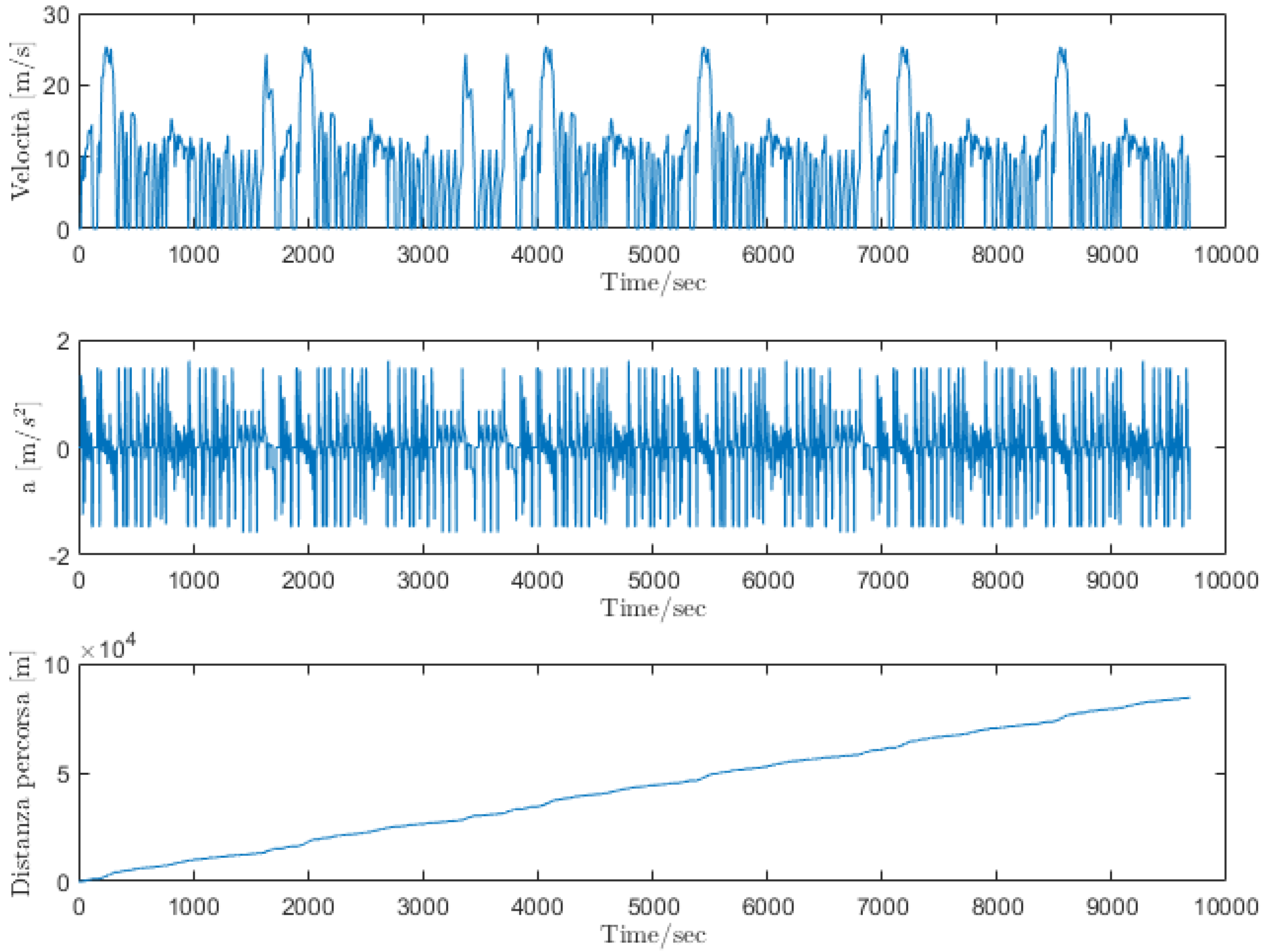
$$T_{wh} = F_{tr} \cdot r_{wh} \quad \Rightarrow \quad T_{pwt} = \frac{T_{wh}}{i_{fd} i_{gb,i} \eta_{mech}^{sgn(T_{wh})}} \quad \Rightarrow \quad \boxed{P_{ice} = \frac{T_{pwt} \cdot \omega_{pwt}}{\eta_{ice}(\omega_{ice}, T_{ice})}, \qquad P_{em} = P_{batt} = \frac{T_{pwt} \cdot \omega_{pwt}}{\eta_{em}^{sgn(T_{pwt})}(\omega_{em}, T_{em})}}$$

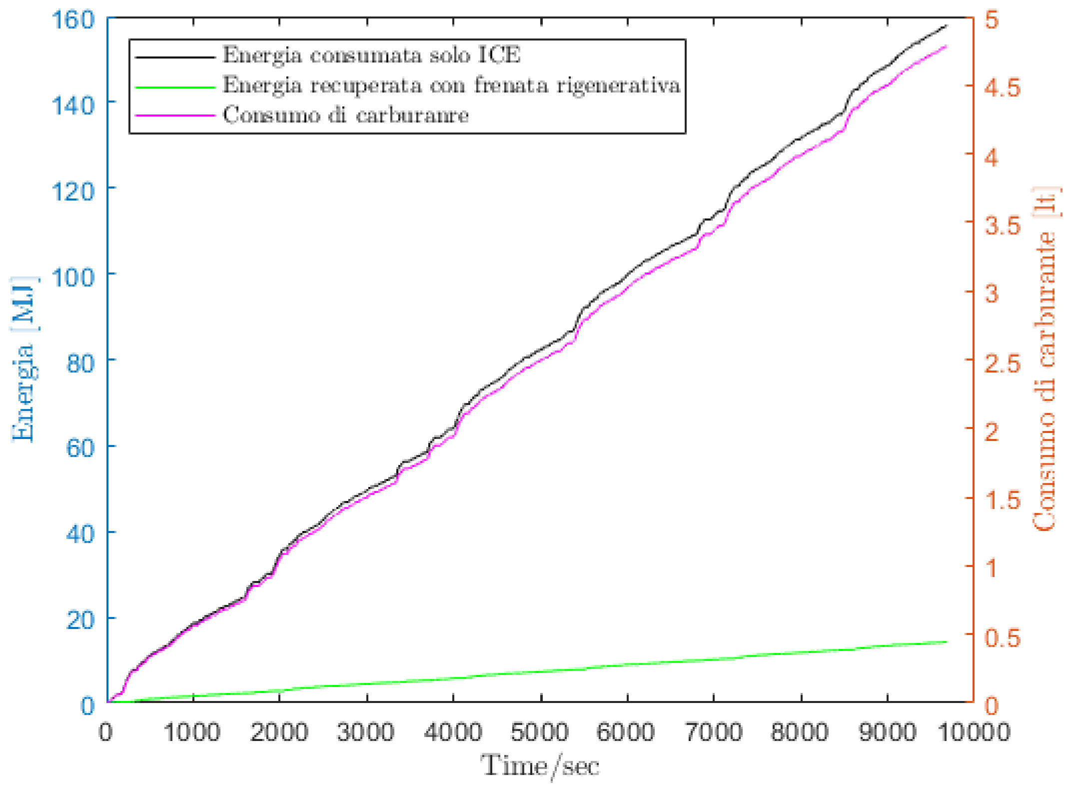
$$\omega_{wh} = \frac{v_{veh}}{r_{wh}} \quad \Rightarrow \quad \omega_{pwt} = \omega_{wh} \cdot i_{fd} i_{gb,i} \quad \Rightarrow \quad \omega_{ice} = \omega_{em} = \omega_{pwt}$$

Spetta alla strategia di controllo bilanciare la potenza richiesta tra ICE ed EM rispettando i vincoli di sistema.

Longitudinal dynamic modelling

— Ciclo di guida urbano (combinazione cicli SFUDS e FUDS)

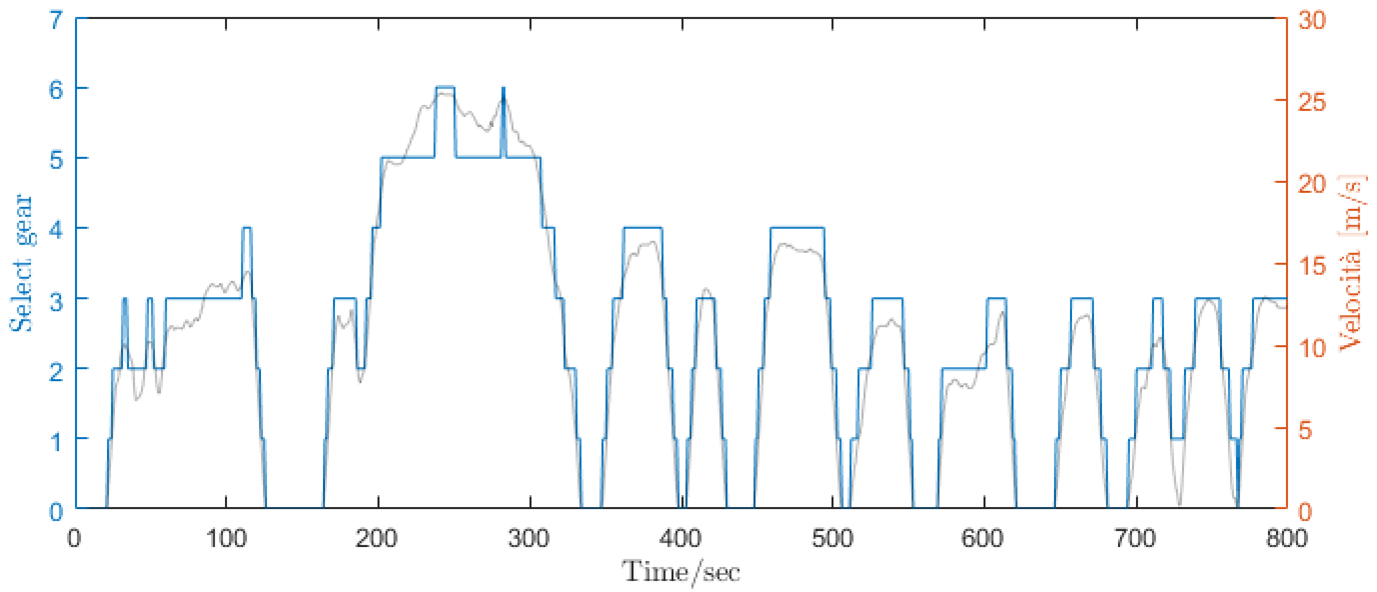




Modello trassmissione

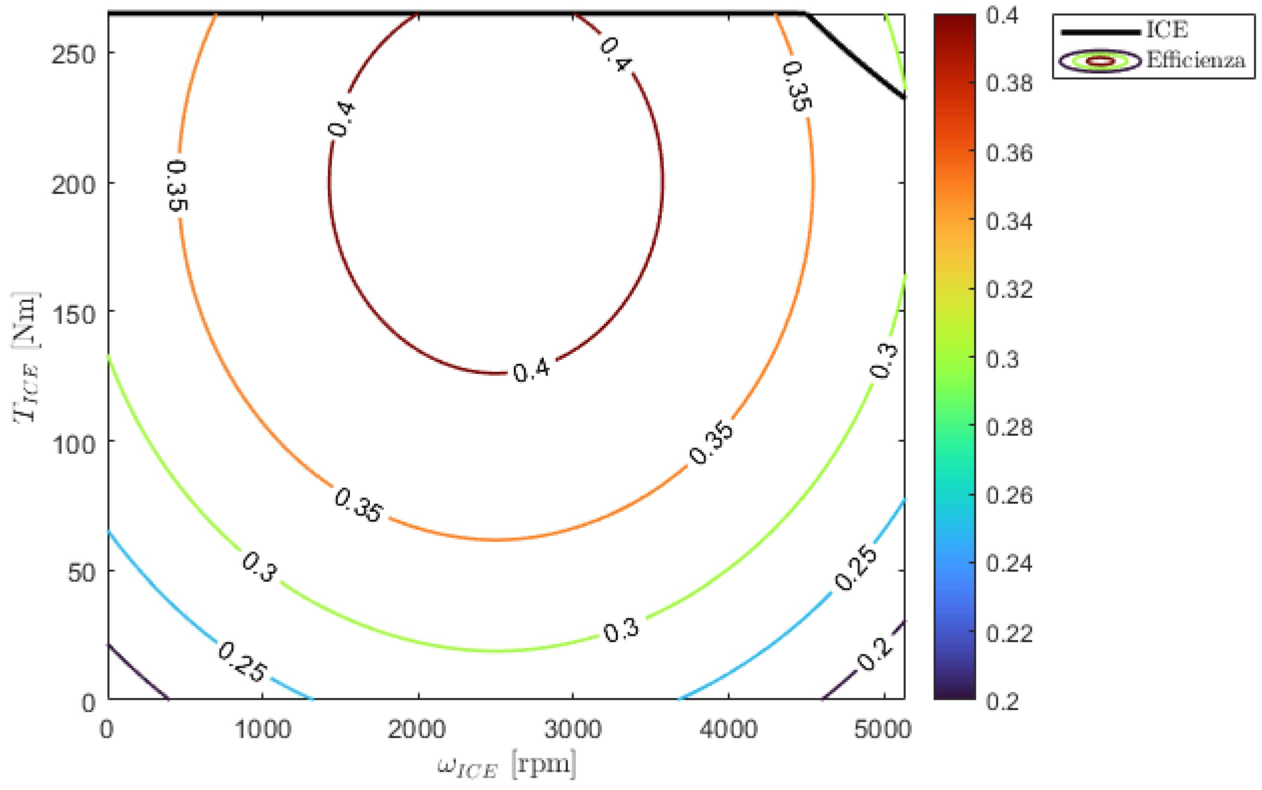
— Modello gear box (Gear ratio in funzione della marcia innestata)

```
function [G,SelGear] = TrasmissionModel(i_fd,i_gb,n_gb,vel_cycle)

G = ones(size(vel_cycle));
SelGear = zeros(size(vel_cycle));
for i=1:numel(vel_cycle)-1
    if vel_cycle(i) > 0 && vel_cycle(i) <= 15/3.6
        SelGear(i) = n_gb(1);
        G(i) = i_fd*i_gb(1);
    elseif vel_cycle(i) > 15/3.6 && vel_cycle(i) <= 35/3.6
        SelGear(i) = n_gb(2);
        G(i) = i_fd*i_gb(2);
    elseif vel_cycle(i) > 35/3.6 && vel_cycle(i) <= 50/3.6
        SelGear(i) = n_gb(3);
        G(i) = i_fd*i_gb(3);
    elseif vel_cycle(i) > 50/3.6 && vel_cycle(i) <= 70/3.6
        SelGear(i) = n_gb(4);
        G(i) = i_fd*i_gb(4);
    elseif vel_cycle(i) > 70/3.6 && vel_cycle(i) <= 90/3.6
        SelGear(i) = n_gb(5);
        G(i) = i_fd*i_gb(5);
    elseif vel_cycle(i) > 90/3.6
        SelGear(i) = n_gb(6);
        G(i) = i_fd*i_gb(6);
    else
        SelGear(i) = 0;
        G(i) = 1;
    end
end

end
```



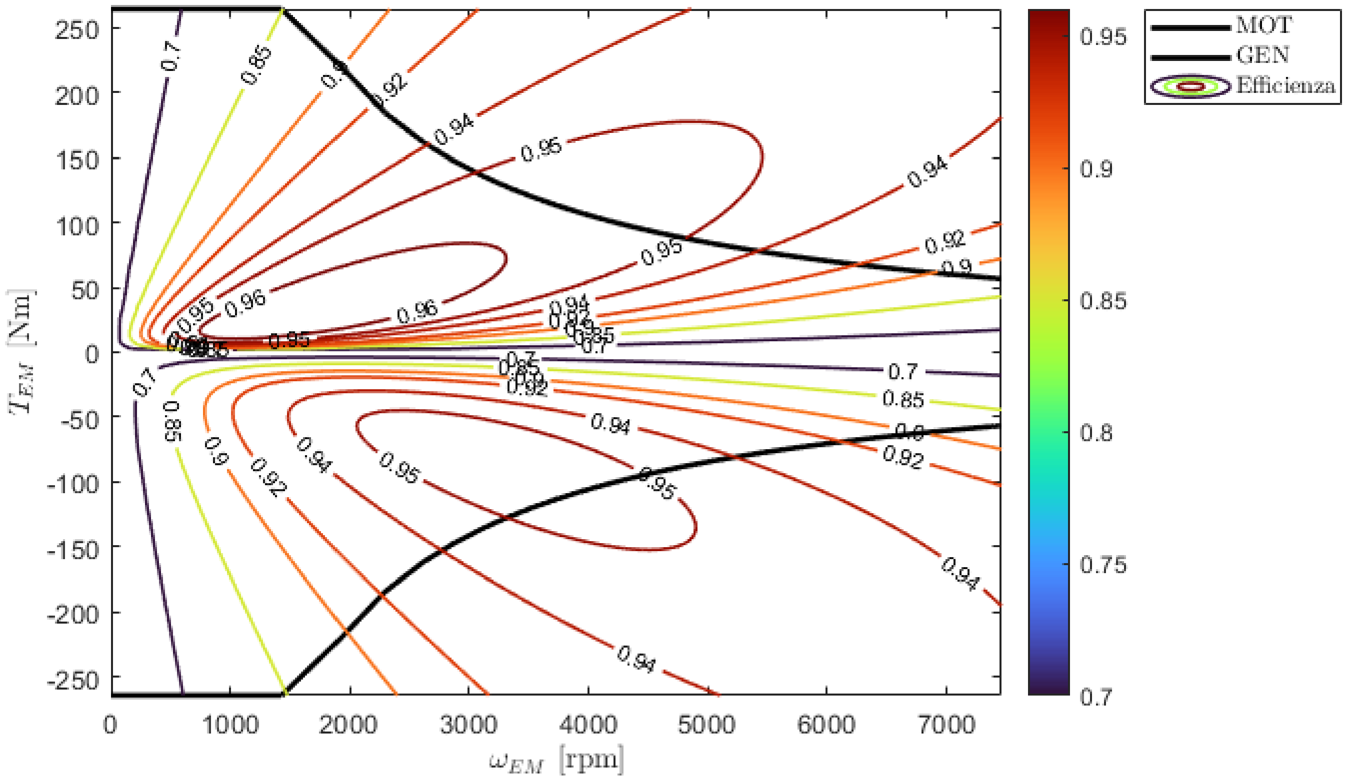
Modello motore a combustione

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% Mappa di efficienza ICE
%%%%%%%%%%%%%%%%%%%%%%%%%

function [val,W_ICE,T_ICE,eta_ICE_map,eta_ICE]=...
    EfficencyICE(w_limite_ICE,T_max_ICE,w_pwt,T_pwt)

W_ICE = linspace(min(w_limite_ICE),max(w_limite_ICE),100);
T_ICE = linspace(0,T_max_ICE,100);
[W_ICE,T_ICE]= meshgrid(W_ICE,T_ICE); % Griglia di punti sul piano
                                      % coppia-velocità angolare

W_rad = (W_ICE*2*pi)/60; % Velocità angolare espressa in [rad/sec]

eta_ICE_map = 0.42-2*10^-6.*abs(W_rad-2500/60*2*pi).^1.95-...
    +0.02/50^2.2.*abs(T_ICE-200).^2;

val = [0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.42,0.44];

% Calcolo efficienza ICE sui punti di lavoro
eta_ICE = 0.42-2*10^-6.*abs(w_pwt-2300/60*2*pi).^1.95-...
    +0.02/50^2.2.*abs(T_pwt-200).^2;

end
```



Modello motore elettrico

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mappa di efficienza MOT e GEN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Val,W,T,eta_MOT_map,eta_GEN_map,eta_EM]=...
    EfficencyElectricMotor(T_max_EM,w_limite_MOT,w_pwt,T_pwt)

kc=0.1; % Coefficiente perdite rame
ki=0.5e-2; % Coefficiente perdite metallo
kw=1.2*10^-5; % Coefficiente perdite aerodinamiche
c_EM=20;
c_GEN=200;

W = linspace(min(w_limite_MOT),max(w_limite_MOT),100);
T = linspace(0,T_max_EM,100);
[W,T]= meshgrid(W,T); % Griglia di punti sul piano coppia-velocità angolare
W_rad = (W*2*pi)/60; % Velocità angolare espressa in [rad/sec]
Out_power = W_rad.*T;
Loss_power_EM= kc*T.^2 + ki*W_rad + kw*W_rad.^3 + c_EM;
Loss_power_GEN= kc*T.^2 + ki*W_rad + kw*W_rad.^3 + c_GEN;
eta_MOT_map = Out_power./(Out_power+Loss_power_EM);
eta_GEN_map = Out_power./(Out_power+Loss_power_GEN);
Val = [0.70, 0.85, 0.90, 0.92, 0.94, 0.95, 0.96];

% Calcolo efficienza MOT e GEN sui punti di lavoro
OutPower = w_pwt.*abs(T_pwt);
TotPowerEM = OutPower + kc*T_pwt.^2 + ki*w_pwt + kw*w_pwt.^3 + c_EM;
TotPowerGEN = OutPower - kc*T_pwt.^2 - ki*w_pwt - kw*w_pwt.^3 - c_GEN;

for i=1:numel(T_pwt)
    if T_pwt(i) >= 0
        % Efficienza motore elettrico
        eta_EM(i) = OutPower(i)/TotPowerEM(i);
    else
        % Efficienza generatore
        eta_EM(i) = max([0,TotPowerGEN(i)/OutPower(i)]);
    end
end

end
```

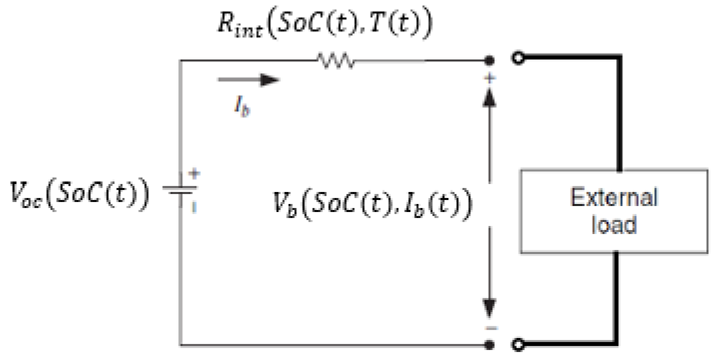


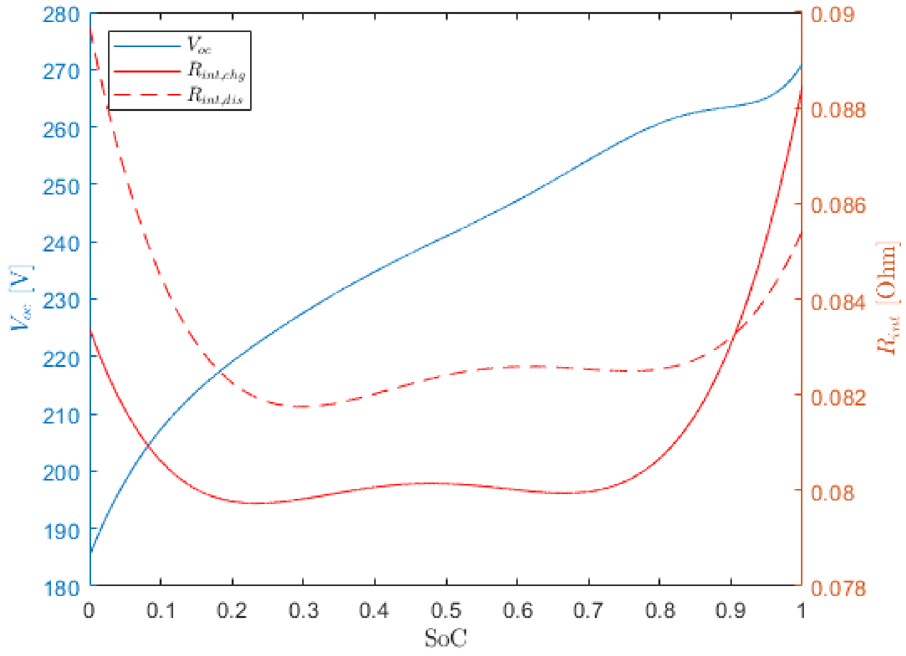
Modello batteria

```
%%%%%%%%%%%%%%%%%%
% Modello batteria
%%%%%%%%%%%%%%%%%%

% Viene utilizzato un modello circuitale equivalente 0th-order per tener
% conto delle perdite resistive (effetto Joule).
% Si determinano Voc e R_int in funzione del SoC attraverso l'utilizzo di
% formule empiriche ricavate sperimentalmente supposto T=25°C.

function [SoC,Voc_comb,Voc_8th,R_int_d,R_int_c,Cp,I_batt]=...
    BatteryModel(t_cycle)

Parametri_Kia_Sportage;

% Stato di carica batteria
SoC = linspace(0,1,length(t_cycle));

% Tensione a vuoto (8th order)
Voc_8th = 65*(k_8th(1)+k_8th(2)*SoC+k_8th(3)*SoC.^2+k_8th(4)*SoC.^3+...
    +k_8th(5)*SoC.^4+k_8th(6)*SoC.^5+k_8th(7)*SoC.^6+k_8th(8)*SoC.^7+...
    +k_8th(9)*SoC.^8);

% Tensione a vuoto (esponenziale + polinomio)
Voc_comb = 63*(a(1)*exp(a(2)*SoC)+a(3)*SoC+a(4)*SoC.^2+a(5)*SoC.^3+a(6));

% Resistenza interna in fase di carica e scarica
R_int_d = (b_di(1)*SoC.^4+b_di(2)*SoC.^3+b_di(3)*SoC.^2+b_di(4)*SoC+...
    +b_di(5))*b_di(6)*exp(b_di(7)/(Te-b_di(8)));

R_int_c = (b_ch(1)*SoC.^4+b_ch(2)*SoC.^3+b_ch(3)*SoC.^2+b_ch(4)*SoC+...
    b_ch(5))*b_ch(6)*exp(b_ch(7)/(Te-b_ch(8)));
```

⋮

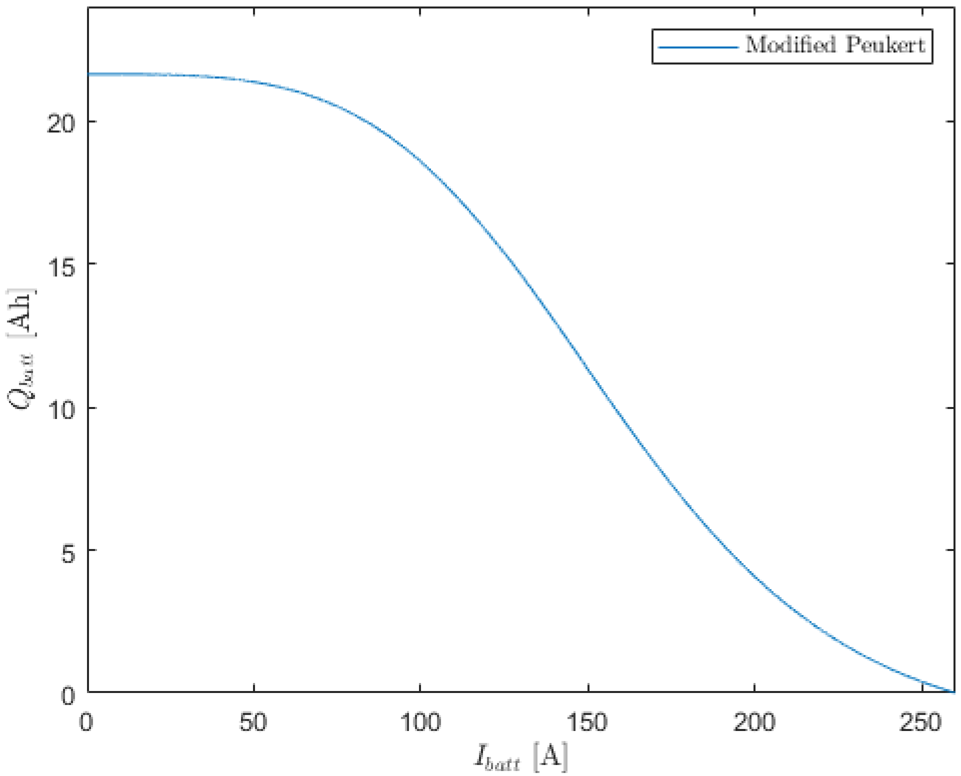
⋮





Modello batteria

```
% Modello Peukert modificato per tener conto degli effetti Coulombiani
I_batt = linspace(0,260,length(t_cycle));
for i=1:numel(I_batt)
    Qp(i) = Qm*(1-I_batt(i)/I1)/((1-I_batt(i)/I1)+(I_batt(i)/I0)^n);
end
```



— Equazione di Peukert generalizzata (Galushkin)

- Rispetto alla classica equazione risolve il problema legato all'utilizzo di correnti molto basse per le quali la capacità tende ad essere infinita.

$$Q_{dis}(I_b(t)) = \frac{Q_m}{1 + \left(\frac{I_b(t)}{i_0}\right)^n}$$

- Svantaggi: non garantisce una capacità nulla alla specifica corrente massima i_1 per la quale la tensione ai capi della batteria risulta inferiore alla tensione di cut-off. Ciò ha portato ad una terza equazione, nota come equazione di Peukert modificata.

— Equazione di Peukert modificata

$$Q_{dis}(I_b(t)) = \frac{Q_m \cdot \left(1 - \frac{I_b(t)}{i_1}\right)}{\left(1 - \frac{I_b(t)}{i_1}\right) + \left(\frac{I_b(t)}{i_0}\right)^n}$$

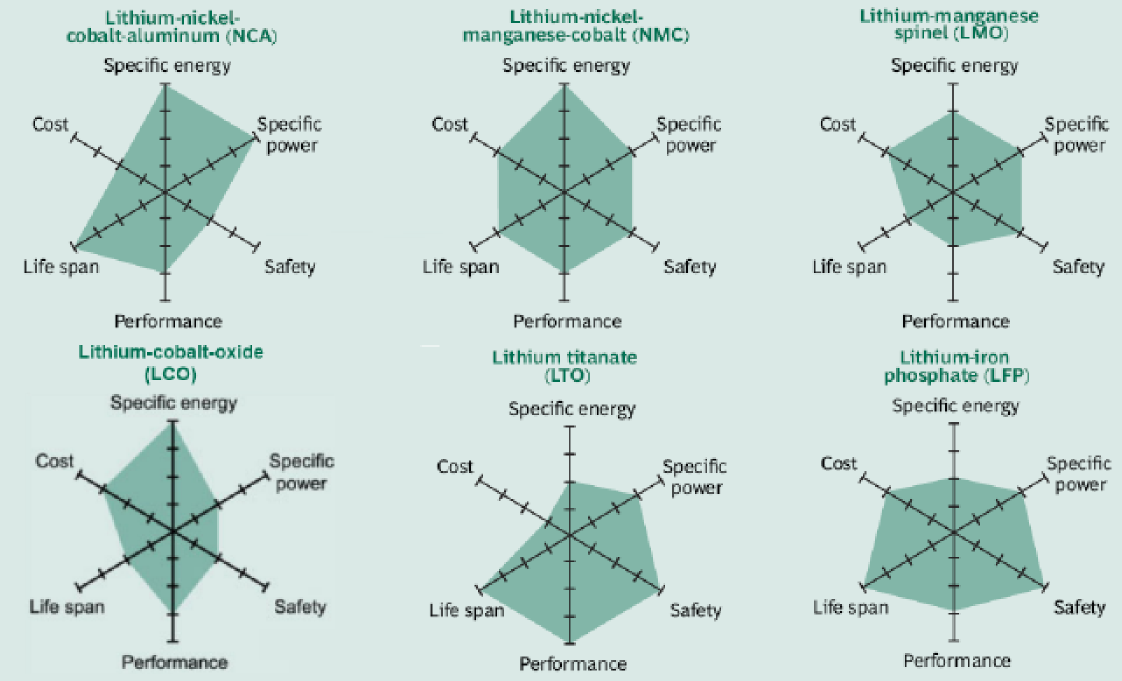
Modello batteria

La corrente può essere modellata attraverso la seguente equazione:

$$I_b(t) = \frac{V_{oc}(SoC(t))}{2R_{int}(SoC(t))} - \sqrt{\left(\frac{V_{oc}(SoC(t))}{2R_{int}(SoC(t))}\right)^2 - \frac{P_{batt}}{R_{int}(SoC(t))}}$$

$$I_b(t) > 0 \ (scarica)$$

$$I_b(t) < 0 \ (carica)$$



Il SoC può essere trovato noto Q_{nom}:

$$SoC(t) = \frac{Q(t)}{Q_{nom}} = \frac{E(t)}{E_{nom}}$$

$$\dot{SoC}(t) = -\frac{I_b(t)}{\eta_{coul}^{sgn(I_b(t))} Q_{nom}} = -\frac{P_{batt}(t)}{\eta_{coul}^{sgn(I_b(t))} E_{nom}}$$

$$SoC(t+1) = SoC(t) - \int_t^{t+1} \frac{I_b(t)}{\eta_{coul}^{sgn(I_b(t))} Q_{nom}} dt$$ **T.C.**

$$SoC(k+1) = SoC(k) - \frac{I_b(k)\Delta t}{3600 \cdot \eta_{coul}^{sgn(I_b(k))} Q_{nom}}$$ **T.D.**

Strategie di controllo

1. Pontryagin's Minimum Principle

Obiettivo della strategia di controllo: trovare il controllo ottimo, $u^*(t)$ con $t \in [t_o, t_f]$, che minimizzi la funzione costo o Hamiltoniana, H, su un orizzonte finito soddisfacendo al contempo i vincoli di stato e co-stato dinamici, il vincolo di stato globale e i vincoli di controllo e di stato locali (problema di ottimizzazione vincolato).

$$u^*(t) = \min_{u(t)\in U(t)} H(u(t), x(t), \lambda(t), t) = \dot{m}_f(t) + \big(\lambda(t) + w(t)\big)\dot{x}(t) \quad \Rightarrow \quad u^* = \min_{u\in U} J = \int_{t_0}^{t_f} \dot{m}_f(t)dt$$

$$x(t) = SoC(t)$$

$$\dot{x}(t) = f(u(t), x(t), t)$$

$$u(t) = P_{EM}(t)$$

$$\dot{m}_f(t) = L(u(t), x(t), t)$$

$$U(t) = [P_{EM,min}, P_{EM,max}]$$

- Condizioni necessarie per l'ottimalitá:

$$\begin{cases} H(u^*(t), x^*(t), \lambda^*(t), t) \le H(u(t), x^*(t), \lambda^*(t), t) \\ \dot{\lambda}^*(t) = -\dfrac{\partial H(u^*(t), x^*(t), \lambda^*(t), t)}{\partial x} \\ S\dot{o}C^*(t) = \dfrac{\partial H(u^*(t), x^*(t), \lambda^*(t), t)}{\partial \lambda} \\ SoC^*(t_0) = SoC(t_0) \\ SoC^*(t_f) = SoC_{target} \\ SoC_{min} \le SoC^*(t) \le SoC_{max} \end{cases}$$

Esiste almeno una traiettoria della variabile di stato che soddisfa i vincoli di sistema e questa costituisce la soluzione ottima globale per la quale la funzione costo globale risulta minimizzata.

$$w(x(t), t) = \begin{cases} K & se \;\; SoC(t) > SoC_{max} \\ -K & se \;\; SoC(t) > SoC_{max} \\ 0 & altrimenti \end{cases}$$

Strategie di controllo

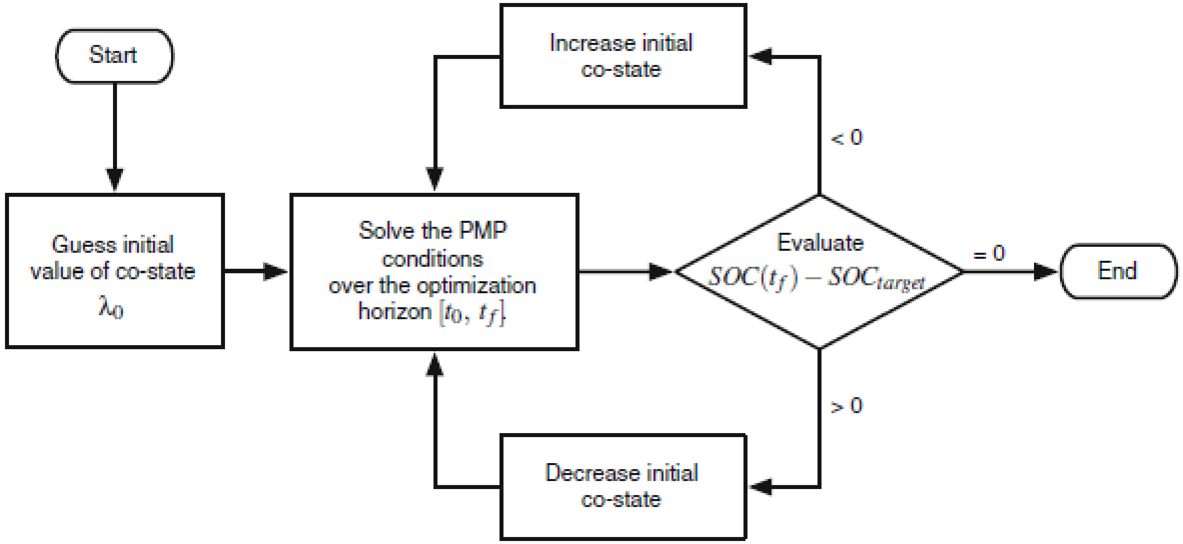
La soluzione PMP può essere espressa in termini di potenza equivalente:

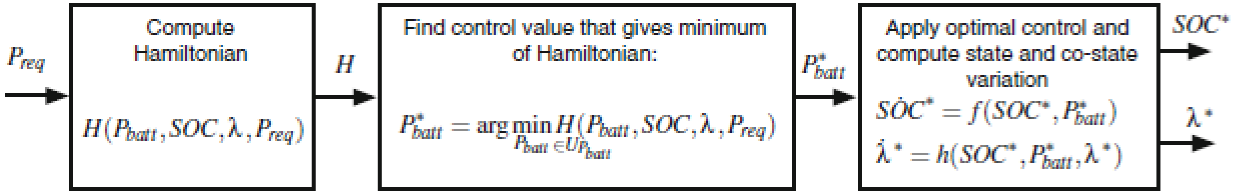
$$H = P_{fuel}(t) + \big(\lambda(t) + w(SoC)\big) \cdot P_{ech}(t)$$

Evoluzione della variabile di stato e del co-stato:

$$S\dot{o}C(t) = -\frac{\partial P_{ech}}{\partial E_{ech}}$$

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial E_{ech}} = -\big(\lambda(t) + w(SoC)\big)\frac{\partial P_{ech}}{\partial E_{ech}}$$

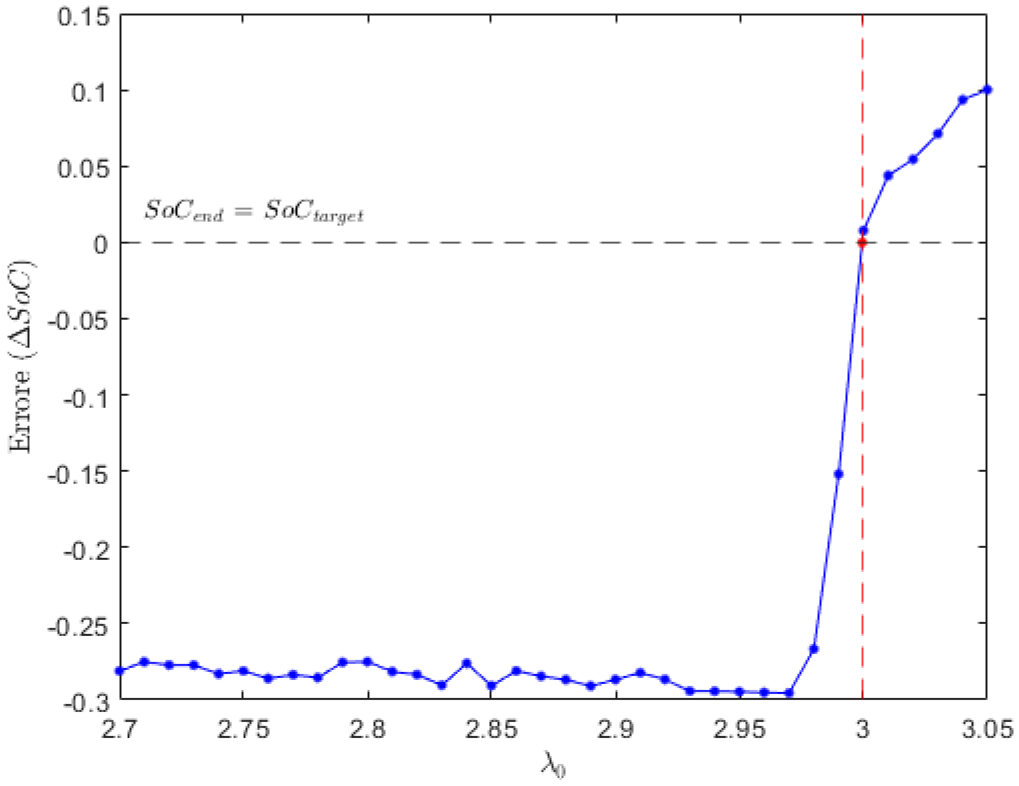


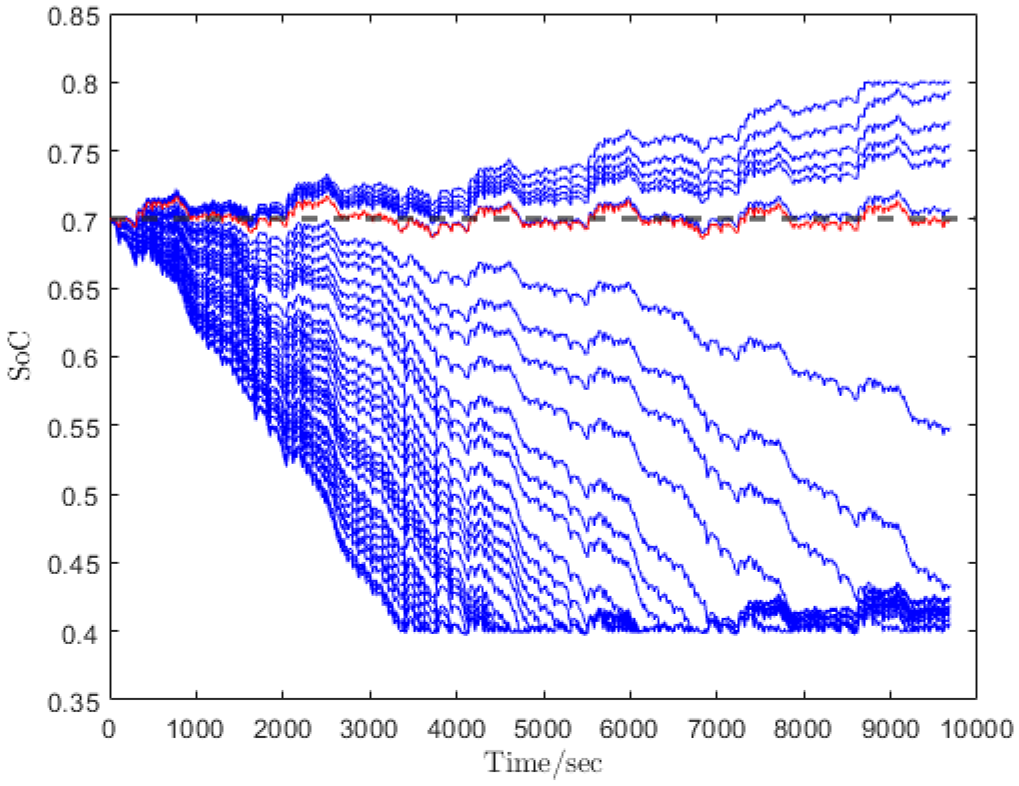


La traiettoria ottimale è molto sensibile al valore iniziale del co-stato λ_0. Nell'implementazione offline, questo viene trovato con una procedura iterativa (Shooting Method). Inoltre, si sfruttata una procedura di bisezione per ottenere la convergenza di λ_0 in poche iterazioni.

Strategie di controllo

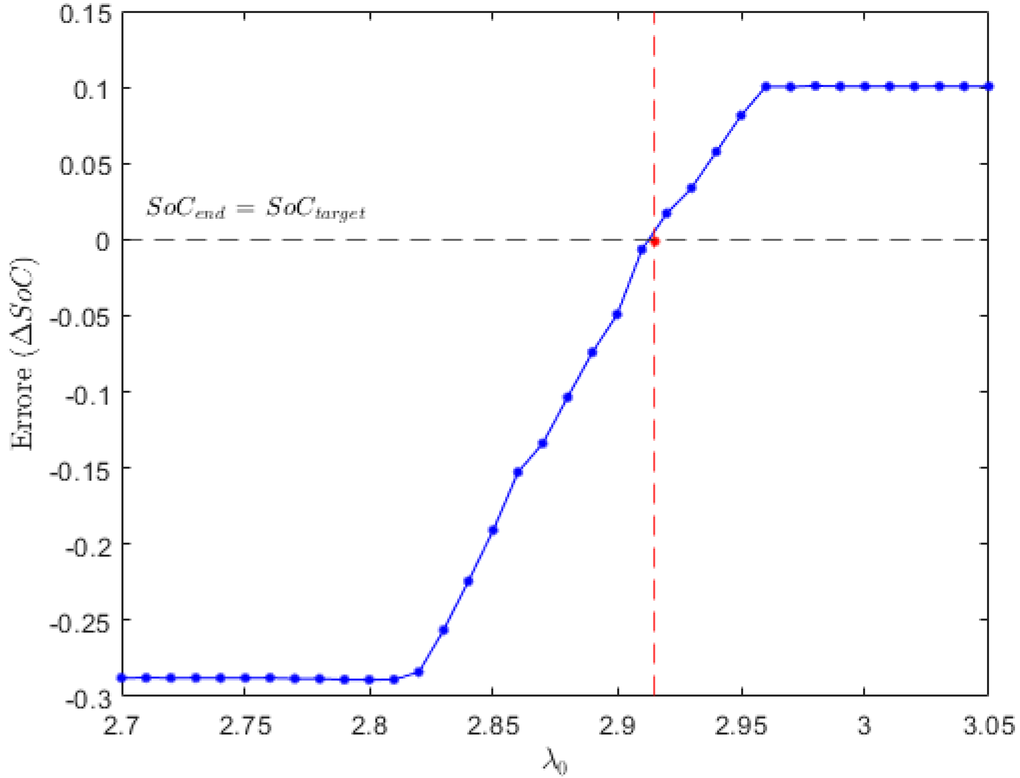
– Ciclo di guida urbano

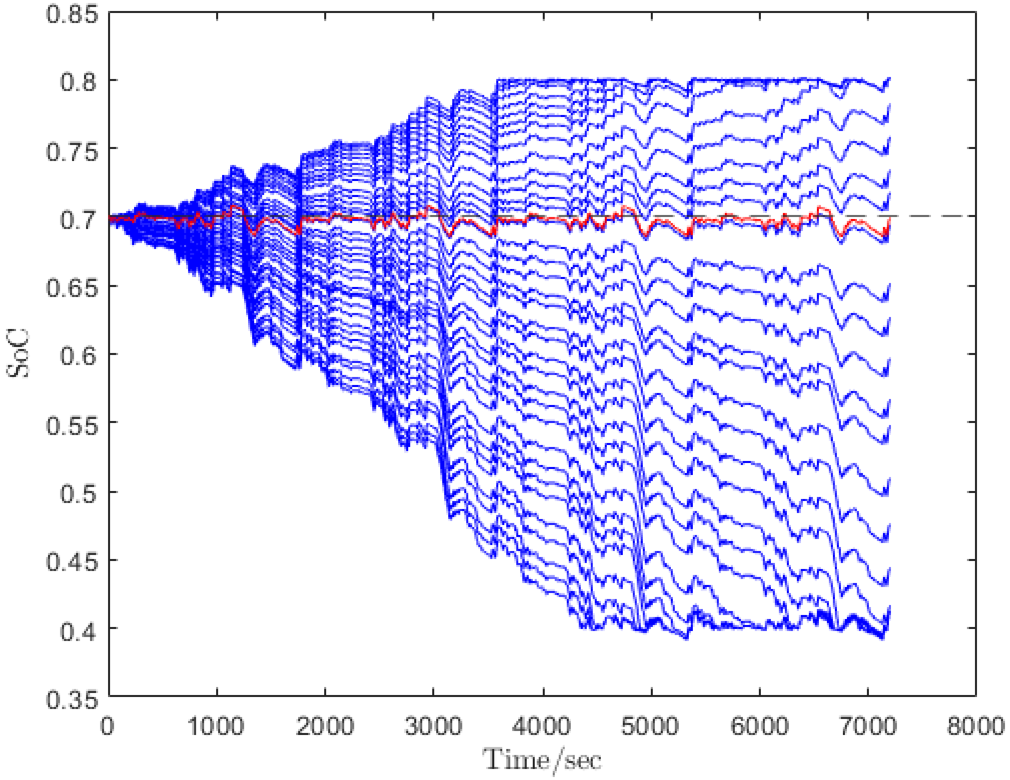




Strategie di controllo

– Ciclo di guida extra-urbano

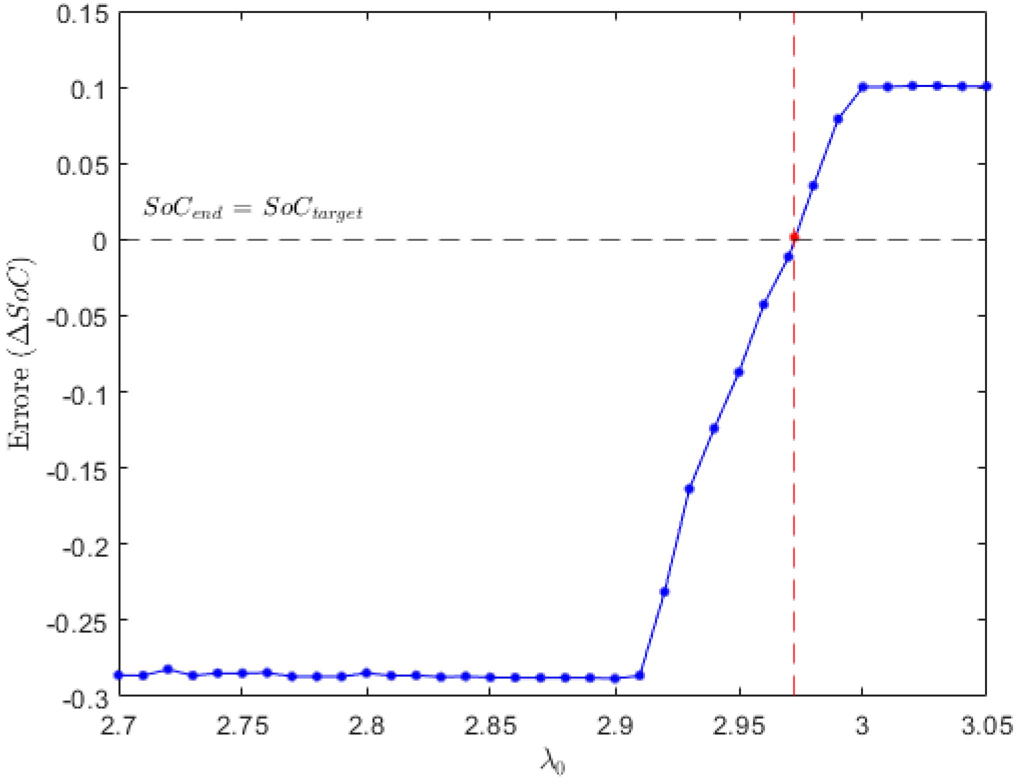


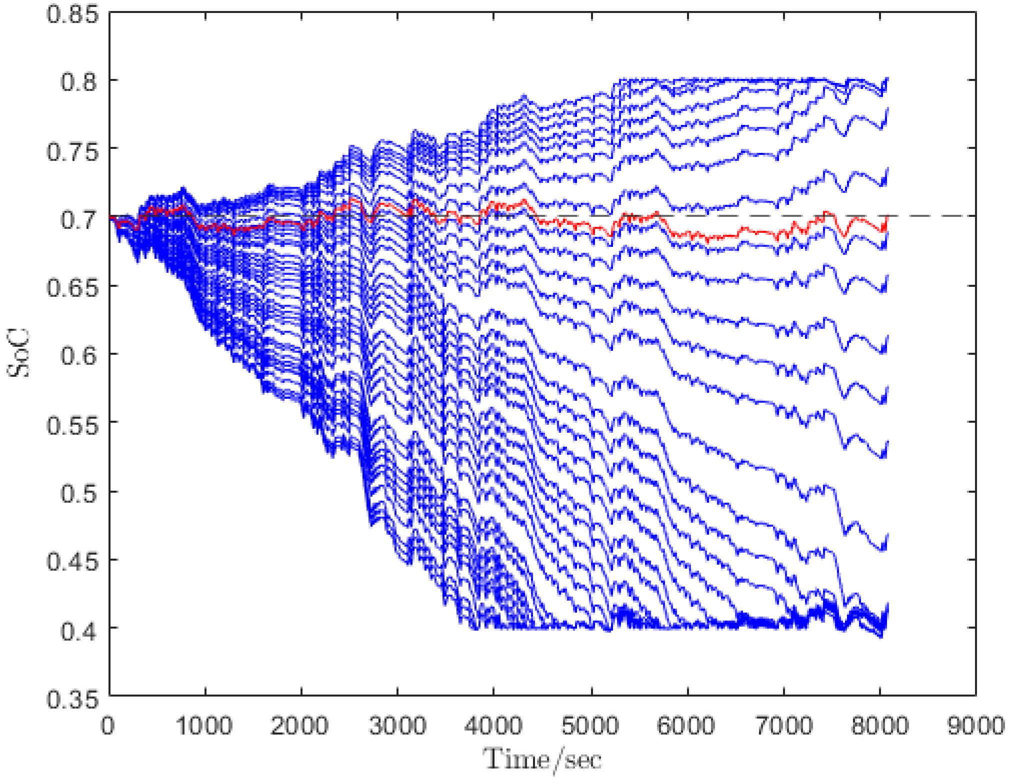


Il co-stato λ_0 ottimale differisce in funzione del ciclo di guida

Strategie di controllo

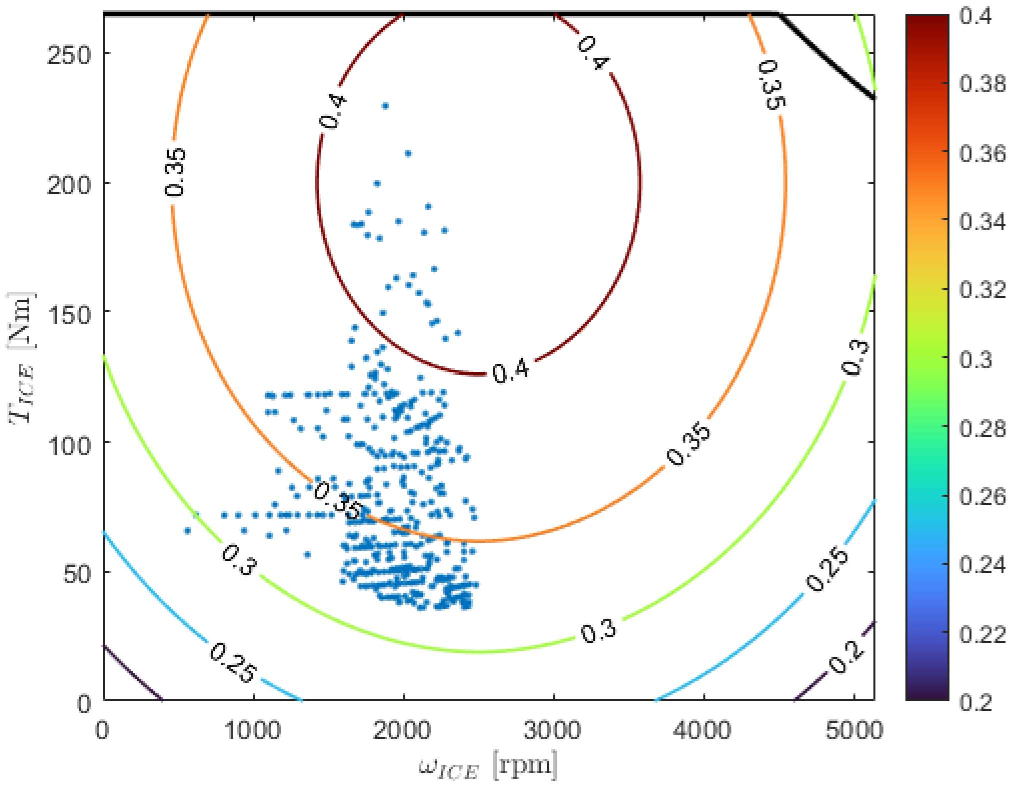
– Ciclo di guida combinato

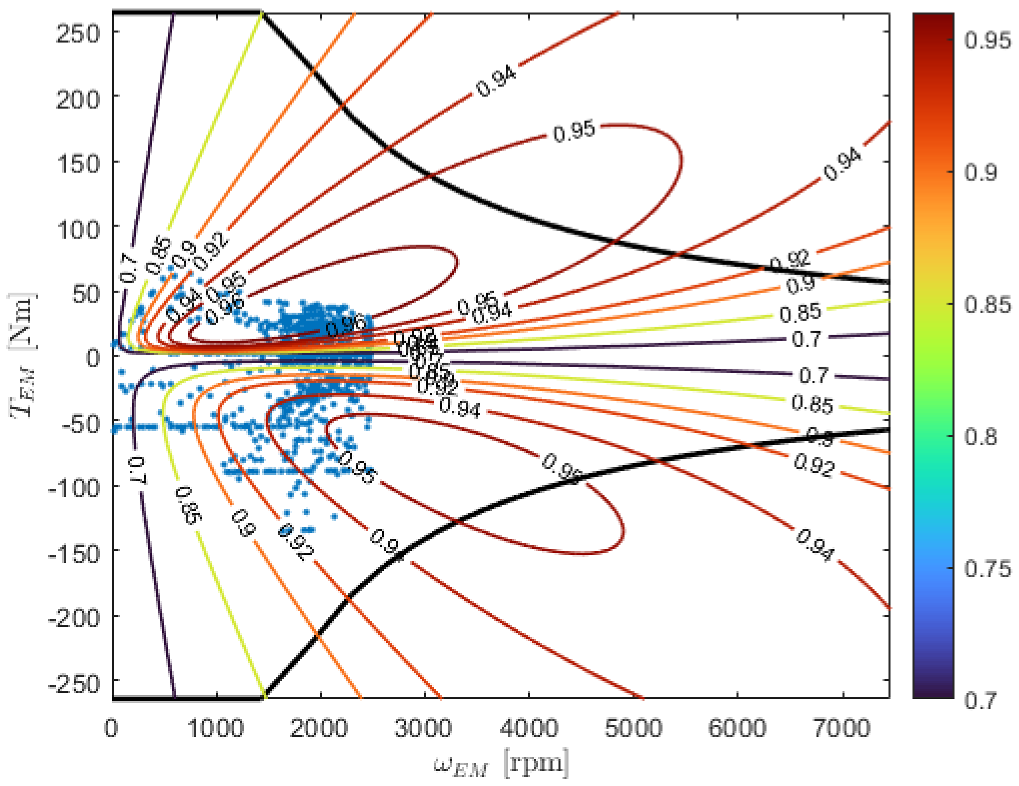




Strategie di controllo

– Punti operativi ricavati dalla strategia di controllo PMP





Strategie di controllo

2. Equivalent Consumption Minimum Strategy (ECMS)

Idea chiave ECMS: un consumo equivalente di carburante può essere associato all'uso di energia elettrica fornita dalla batteria sia in carica che in scarica e sommato al consumo reale di carburante per ottenere il consumo equivalente istantaneo di carburante.

$$\dot{m}_{f,eqv}(t) = \dot{m}_f(t) + \dot{m}_{ress}(t)$$

– Sviluppata sulla base della strategia PMP:

$$H = P_{fuel}(t) + \big(\lambda(t) + w(SoC)\big) \cdot P_{ech}(t) \quad \longrightarrow \quad P_{eqv}(t) = P_{fuel}(t) + s_{chg,dis}(t) \cdot P_{batt}(t) \cdot p(SoC)$$

$$s_{chg}(t) = \lambda(t)\eta_{batt}$$

$$s_{dis}(t) = \frac{\lambda(t)}{\eta_{batt}}$$

– La sua equivalenza con la strategia PMP consente di generare una soluzione ottimale

Strategie di controllo

Note su PMP ed ECMS:

- Non adatti all'implementazione real-time a causa della loro natura non causale
- Implementabili solo nell'ambito della simulazione numerica (offline)
- Utili per progettare regole per l'implementazione online e/o come soluzione di riferimento per valutare le prestazioni di altre strategie di controllo

3. Continuous Adaptive-ECMS e Discrete Adaptive-ECMS

- Strategie di natura causale basate su regole, quindi implementabili in real-time (online)
- Nell'implementazione online il fattore di equivalenza non è noto a priori
- Si Sfrutta, ad esempio, l'adattamento tramite il feedback del SoC per adattare dinamicamente in ogni istante, o periodicamente, il fattore di equivalenza generando una soluzione sub-ottimale
- Data l'equivalenza con PMP, l'adattamento può essere rivolto a un solo parametro, es. il co-stato, poiché l'efficienza è implicitamente presa in considerazione nella valutazione di P_{ech}.

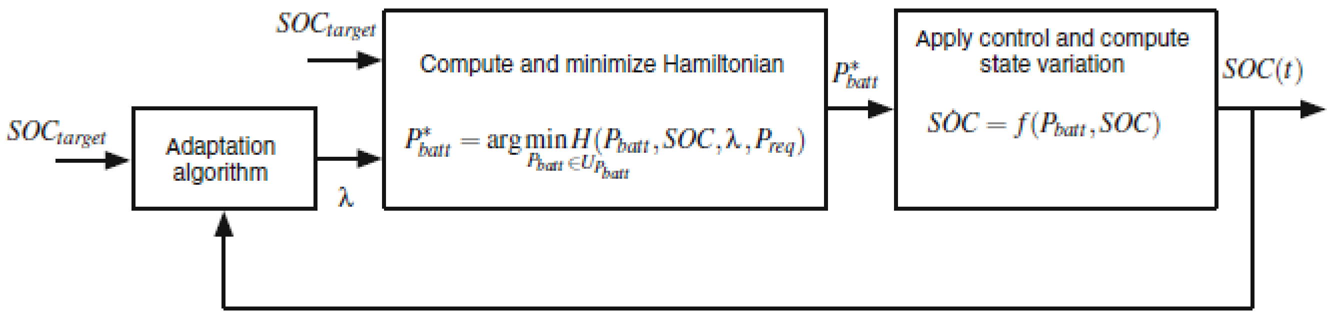
Strategie di controllo

– **Continuous A-ECMS**

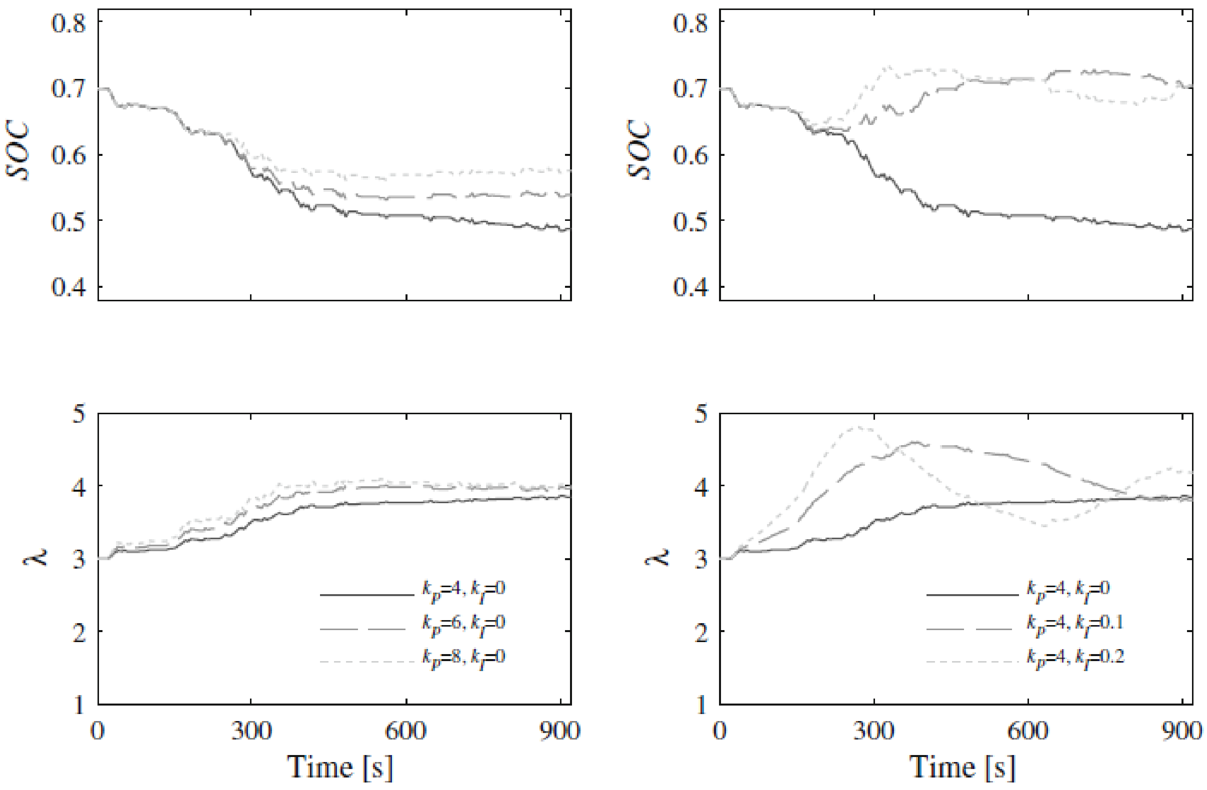
$$\lambda(t) = \lambda_0 + k_P \left(SOC_{target} - SOC(t)\right) + k_I \int_0^t \left(SOC_{target} - SOC(\tau)\right) d\tau$$

– **Discrete A-ECMS**

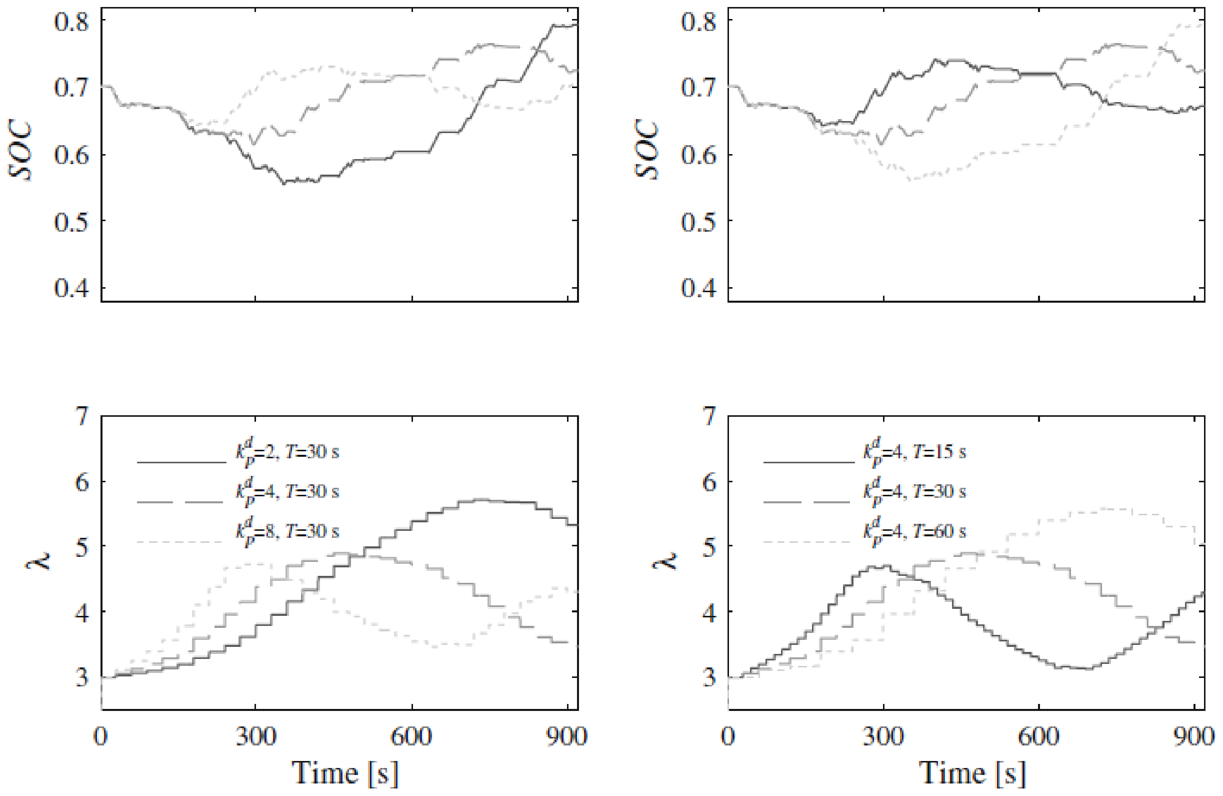
$$\lambda(k) = \frac{\lambda(k-1) + \lambda(k-2)}{2} + k_P^d \left(SOC_{target} - SOC(k)\right)$$



Strategie di controllo



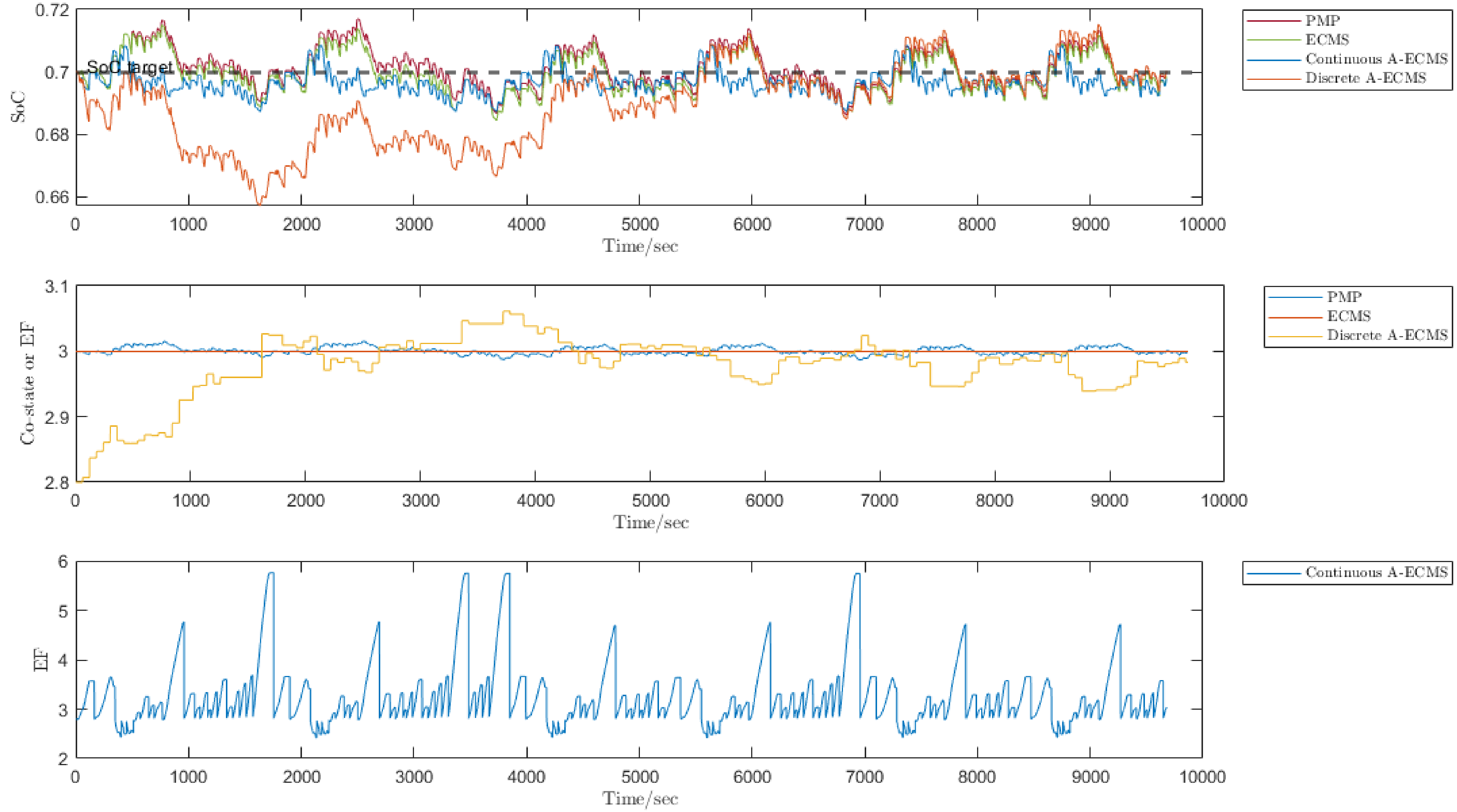
Continuous A-ECMS



Discrete A-ECMS

Strategie di controllo

– Confronto strategie di controllo su ciclo urbano



$$S_{ECMS,dis}(t) = \lambda_0^*/\eta_{batt,avg}$$

$$S_{ECMS,chg}(t) = \lambda_0^* \cdot \eta_{batt,avg}$$

$$\lambda_{DA-ECMS}(0) = \lambda_0^* - 0.2$$

$$K_P^d = 3$$

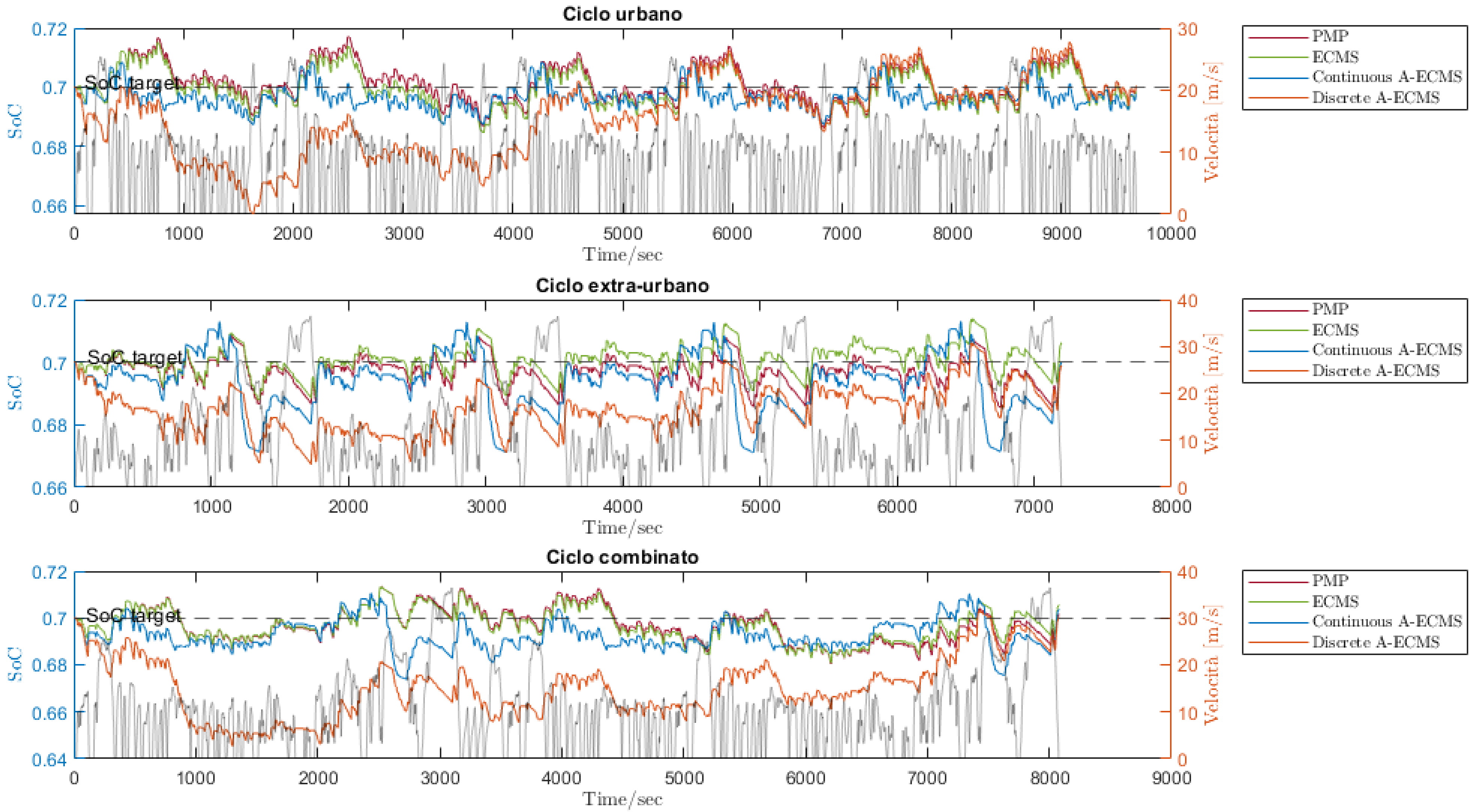
$$T = 60\ s$$

$$\lambda_{CA-ECMS}(0) = \lambda_0^* - 0.2$$

$$K_P = 4$$

$$K_I = 2$$

Conclusioni



Conclusioni